Secure Web App Handoff

Andrew Diana

Secure Web App Development

App: 2p Snake.io

Web App URL: <https://safe-dusk-69192.herokuapp.com>  
Backend URL: <https://shrouded-eyrie-80233.herokuapp.com/>

Database: <https://mlab.com/home>

Username: JavaScript

Password: Professor123!

Frontend / Backend: <https://id.heroku.com/login>

Front End (Vue + Node File Serving): <https://safe-dusk-69192.herokuapp.com>

Back End (Node + Express + WebSocket): <https://shrouded-eyrie-80233.herokuapp.com/>

I added  [dsbics@rit.edu](mailto:dsbics@rit.edu) as a collaborator on Frontend and Backend

About the app- if you are going to log into the app- please use different browsers / incognito mode.

If the app doesn’t log in immediately- it has to spin up the frontend and the backend on heroku which takes about 30 seconds.

Let me know if there is anything else that you need.

My Phone #: (516) 491-1348

Email: [axd7832@rit.edu](mailto:axd7832@rit.edu)

### 

### Frontend

*final\_app/Frontend/final-project/src*

In here is where all the front end Vue stuff is going on. The front end basically works as such:

User Login -> Token send to the client -> token attached to all WebSocket requests

The game first puts all users into “Room 0” which is the game lobby. Players can chat in this game lobby and all of the other players in the room will receive the messages.

From here there is a button on the top right of the screen (or in the dropdown) to invite another player. This will show you a list of all of the players (not including you) that are currently online. Click “Invite” to send the request. You should then see in the chat that an invite has been sent.

\*on the other client you can accept/deny the request.

The response of that accept/deny is sent to the user that sent the invite.

Once the invite has been accepted: you will be put into a new lobby with the other player.

You should now see a screen saying “Waiting for both players to Ready Up”.

Clicking the Ready Up button will change the text to “Awaiting Other Player” as well as sending a chat from the server “You have readied up.”

The game should now send a chat from the server alerting you that the game has been started saying “The game has started. Take control of the snake with WASD”  
  
You can now send commands with WASD to move the snake.

The game is a bit wonky… it’s a damn shame that you are not able to fully experience the comedy of two players fighting to control the snake in a direction.

/components are where the individual pages are located

The Game component is where the game logic for the frontend is located

.

### Back End

*final\_app/Server*

*/Server.js*

This file is the point of entry of the server. The server connects to the database and emits an event once the database is connected with Mongoose.

This file also adds security such as *helmet.js* and string validation *express-mongo-sanitize*

The api routes are added with the import of the *./routes* functions

The WebSocket is started here as well.

*/routes/index.js*

This file handles the register and login functionality.  
The register post takes the username and password from the req body and cleans it and then tries to add it to the db. If the creation is successful, the account is created and a 200 is returned. 400 if not.

The login takes the params that the body of the request, cleans it, and then compares the hash of it to the password hash stored in the db from the requested user. If the hashes match- a token in generated with the user information (username, authToken). This information is then sent to the frontend on a valid login.

*/connections/socketConn.js*

This is the file that houses the WebSocket events.

When the socket receives an event from the clients, it reacts to it and performs an action, and then emits to either the socket or to everyone in a room.

Some of the logic is exported into helper files located at:

*final\_app/Server/game/*

It’s in here these files that move the snake, check the location, create a food for the snake, and handles the invite/ready system.

### Future Updates

* Change the Game logic to use Classes or Prototypical JS classes.
  + (This is mostly due to rushing to get the game logic / board completed)
* Change the SVG game board to Canvas.js for easier painting of the snake.
* Implementing a custom domain ( not heroku’s default domain name )
  + Implementing custom TLS/SSL Cert
* Redirect from http → https